

Setting up Munix V.2/6a

Best.-Nr.: D929508-0686

DF: set06

Author's initials: IW, JH, NR

Trademarks:

MUNIX,	for PCS
DEC, PDP	for DEC
UNIX	for Bell Laboratories

Copyright 1986 by

PCS GmbH, Pfälzer-Wald-Strasse 36, D-8000 München 90, tel. (089) 68004-0

The information contained herein is the property of PCS and shall neither be reproduced in whole or in part without PCS's prior written approval nor be implied to grant any license to make, use or sell equipment manufactured herewith.

PCS reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented.

Table of Contents

1. Introduction	1
2. Getting Started with a Configured System	2
2.1. Loading Unix	2
2.1.1. CADMUS 9230/9240	3
2.1.2. CADMUS 9520	4
2.1.3. CADMUS 9212 Ext	5
2.2. Reading Filesets	6
2.3. Modify Files in /etc	8
2.4. Virtual and Nonvirtual Systems	11
3. Getting Started with an Unconfigured System	12
3.1. Saving Old Software	12
3.1.1. MUNIX 1.4	13
3.1.2. MUNIX 1.5	13
3.2. Coldstart	14
3.2.1. CADMUS 9230/9240	16
3.2.1.1. Check	16
3.2.2. CADMUS 9520	18
3.2.2.1. Check	18
3.2.3. CADMUS 9212 Ext	22
3.2.3.1. Check	22
3.2.4. Mkfs	26
3.2.5. Restor	27
3.3. System Installation	28
3.3.1. Startup MUNIX	28
3.3.1.1. Root and Swap Devices	28
3.3.2. Setting the Correct Date	31
3.3.3. Special Files	31
3.3.3.1. Disk Partitioning and Swap Device	32
3.3.4. Filesystems	32
3.3.4.1. CADMUS 9230/9240	32
3.3.4.2. CADMUS 9520 and CADMUS 9212 Ext	35
3.3.5. Reading Filesets	35
3.3.6. Adapting /etc Files	38

3.4. Loading Unix	42
3.4.1. Installation of the Virtual or Nonvirtual Kernel	42
3.4.2. Loading Unix on Different CADMUS Systems	44
3.4.2.1. CADMUS 9230/9240	44
3.4.2.2. CADMUS 9520	45
3.4.2.3. CADMUS 9212 Ext	46
3.4.3. Restoring	46
4. New Kernel	47
4.1. /etc/newconf	47
4.1.1. Drivers	47
4.1.2. 18 Bit Controllers	48
4.1.2.1. DMA wiring	48
4.1.3. Root and Swap	49
4.1.3.1. CADMUS 9230/9240	49
4.1.3.2. CADMUS 9520 and CADMUS 9212 Ext	50
4.1.4. Questions which are not asked	50
4.2. Generate New Kernels	51
4.3. Virtual and Nonvirtual Systems	51
5. Installing MUNIX V.2/6a as Update to MUNIX V.2/04	53
5.1. Reading Tape or Streamer	54
5.2. Administrative Files	54
5.2.1. Modem Control for Console	55
5.2.2. Terminal Names for CADMUS 9600 and CADMUS 9900	55
5.3. New Kernel	56
5.4. MUNIX/NET Update	57
5.4.1. First Steps	57
5.4.2. Remove MUNIX/NET 1.1	57
5.4.3. Reading Fileset	58
5.4.4. Installation	58
5.5. Finishing Touches	60
5.5.1. fsh	60

* * * *

1. Introduction

Systems by PCS come in a wide range of hardware configurations. This makes the setting up process a nontrivial task. We have changed the format of the setting-up notes considerably in comparison with earlier releases. They now consist largely of five sections. Please note that each section contains information concerning the systems **CADMUS 9230/9240**, **CADMUS 9520** (Tower) and **CADMUS 9212 or 9220 with CADMUS 2802-62 Extension Package**. (For these last two systems the short name **CADMUS 9212 Ext** will be used in this notes.) Where necessary, differences have been included under different headings. It is important to distinguish between these CADMUS models while working your way through the setting-up notes.

Chapter 2 explains how to get a CADMUS system running which has been delivered with the operating system MUNIX V.2/06 already installed. It also contains information about individual system adaptations which the user must carry through himself.

Chapter 3 is for users who will be installing MUNIX V.2/06 from scratch and who, for example, may have already had MUNIX 1.5 running on their systems. It contains the necessary instructions for loading the software from tape or cartridge, adapting the system configuration to the available hardware, setting up filesystems, etc. The individual steps of the setting-up process each constitute a section of the chapter.

Chapter 4 is devoted to MUNIX kernel generation using the procedure /etc/newconf.

Chapter 5 is for users who will be updating from MUNIX V.2/04 to V.2/06. This chapter was written for all PCS CADMUS systems, i.e. not only for the models mentioned above, but also for the **CADMUS 9600** and **CADMUS 9900** systems.

2. Getting Started with a Configured System

In this chapter we will be assuming that MUNIX V.2/06 has already been installed on the disk of your CADMUS system. This means a **usr** filesystem and a **root** filesystem which contains a kernel specially configured for your system is available.

The following sections describe how to bring up the system, how to read in other software from magtape or streamer and how to make specific system modifications.

2.1. Loading Unix

Your root filesystem contains several unix kernels. **/unix**, the virtual kernel **vunix**, and the nonvirtual kernel **nunix**, all configured for your system, and **/aunix**, which can run on a variety of disks and terminals and be used for the coldstart of a CADMUS system.

An order for a nonvirtual system contains unix, nunix and aunix. An order for a virtual system contains all 4 unix kernels with unix being a copy of nunix.

As the minitor versions 2.301, 2.400 and higher have autoboot capability, **/unix** will be loaded automatically after hitting the **INIT** key, or by turning the key to the **BOOT** position. The following sections illustrate the different loading processes on the different CADMUS models.

2.1.1. CADMUS 9230/9240

System	Your Input	Comment
	Hit INIT key	
<i>Minitor prompts with</i> <i>Minitor 2.3xx</i> <i>Ethernetaddr 08002700xxxx</i> <i>Serialnum xxxx</i> <i>autoboot (hit any key to break)</i>	nothing	these two messages only for Minitor 2.306 a few seconds will pass before the following message appears autoboot
<i>rk0/unix</i> ... <i>INIT:SINGLE USER</i> <i>going Multi-user mode? (y/n):</i> y <i>Is the date ... correct? (y or n)</i> y		System information Correct dates are entered: mmddhhmmmyy, (month first)
<i>Do you want to check</i> <i>the filesystems? (y or n)</i> ... <i>pcs (this is V.2/06)</i> <i>Console Login:</i>	y fsck information root	

2.1.2. CADMUS 9520

System	Your Input	Comment
	Switch key from position ON to position BOOT, release key	
<i>Minitor prompts with Minitor 2.4xx autoboot (hit any key to break)</i>	nothing	a few seconds will pass before the following message appears autoboot
<i>rc0/unix</i>		System information
<i>... INIT:SINGLE USER going Multi-user mode? (y/n): Is the date ... correct? (y or n)</i>	y y	Correct dates are entered mmddhhmmmyy, (month first)
<i>Do you want to check the filesystems? (y or n) ... pcs (this is V.2/06) Console Login:</i>	y fsck information root	

2.1.3. CADMUS 9212 Ext

System	Your Input	Comment
<i>Minitor prompts with Minitor 2.4xx autoboot (hit any key to break)</i>	Hit INIT key nothing	a few seconds will pass before the following message appears autoboot
<i>rc0/unix</i>		System information
<i>... INIT:SINGLE USER going Multi-user mode? (y/n): Is the date ... correct? (y or n)</i>	y y	Correct dates are entered: mmdhhmmmyy, (month first)
<i>Do you want to check the filesystems? (y or n)</i>	y	fsck information
<i>... pcs (this is V.2/06) Console Login:</i>	root	

2.2. Reading Filesets

You should ascertain which software is already on your system and read in any other that you need, as follows:

The standard distribution of MUNIX V.2 has been grouped into 16 filesets, one for the root filesystem (*rootlist*) and 15 for the *usr* filesystem. This allows the system administrator to tailor the system to the local requirements, i.e. space or usage constraints.

In the directory **/usr/local/filesets** you will find several files ending with '*list*' (e.g. *kernel_vlist*). These files contain the pathnames of the files distributed with your system.

If you want to use files from other filesets (like *man*, *lex*, or *uucp*), you will have to read them in from your distribution media.

The order of the filesets on magtape or cartridge is as follows:

list	size in blocks	size in MB	contents
kernel_v	6858	3.42	all kernel related files for virtual systems
or kernel_s	4333	2.16	all kernel related files for non-virtual systems
usr	4860	2.43	most usr files
terminfo	1756	0.87	screen handling
icc	384	0.19	icc related software
acct	697	0.34	Accounting
games	1813	0.90	Games
graphic	376	0.18	simple graphics
lexyacc	985	0.49	lex, yacc, awk
lp	766	0.38	printer spooler
man	3788	1.89	manual pages
nroff	764	0.38	simple formatting
testmon	890	0.44	testmonitor
tmac	1183	0.59	necessary macros for nroff and troff
ucb	1905	0.95	Berkeley utilities
uucp	809	0.40	UNIX-UNIX Communication

When reading in any fileset, note the following:

- you must be superuser

login root

- don't use the C-Shell but the Bourne Shell (standard shell).

- go to / in all cases:

cd /

- each fileset consists of one distinct cpio file on streamer or magtape.

- when reading in filesets, first rewind the tape or cartridge and then skip the filesets you do not need.

Device	Commands	Comment
<u>CADMUS 9230/9240 and CADMUS 9212 Ext.:</u>		
tape:	/etc/mt rew /etc/mt fsf <i>n</i> cpio -iBmvd < /dev/nrmt0	rewind tape skip <i>n</i> filesets on tape read a fileset without rewinding
streamer:	</dev/rst0 stskip <i>n</i> cpio -iSmvd < /dev/nrst0	rewind streamer cartridge skip <i>n</i> filesets on cartridge read a fileset without rewinding
<u>CADMUS 9520:</u>		
streamer:	</dev/rwt0 wtskip <i>n</i> cpio -iSmvd < /dev/nrwt0	rewind streamer cartridge skip <i>n</i> filesets on cartridge read a fileset without rewinding

Continue by alternating the skip and read commands according to your needs. You may want to write a small shell loop which reads in a number of filesets:

```
for i in 0 1 2 3 4 5 6 7 8 9 10
do
  cpio -iSvmd </dev/nrst0
done
```

The above commands read 11 filesets from the streamer cartridge on a CADMUS 9200 system.

Important:

It is recommended that at least the first 3 filesets exist on your system. The first fileset, named kernel_v or kernel_s, contains all the files which are necessary to make a new kernel, or to switch to the virtual system. The second fileset, usr, contains most files in /usr/bin. It

is also recommended to read in the **terminfo** fileset as it contains the description of terminal characteristics. If you want to use additional utilities like **lex + awk** or **uucp**, you have to read in the corresponding filesets from tape or cartridge.

If you wish to check, save or remove filesets, we suggest you use the shellscripts provided in **/usr/local/filesets** and the lists **/usr/local/filesets/*list**. The entries in this lists have been changed to allow checks for correctness. Each pathname is followed by 2 numbers, the output of the program **sum**. The shell script **ckft** may thus check for existence and consistency of all files belonging to a fileset.

This was done for all filesets belonging to the MUNIX standard distribution, we will do the same for all optional packages in the future.

ckft <fileset> ...

Check a fileset for completeness and consistency on disk.

cpioft <unit> <fileset> ...

Copy a fileset to unit, where unit may be a streamer (st,wt or is), a magtape (mt), or a directory (directory name).

rmft <fileset>

Remove a fileset from disk.

The shell script **/usr/local/copyboot** puts the 2 startup files and a dump of the root filesystem on a streamer or magtape. This shell script may be used for a backup of your root filesystem after you have added all your local additions.

2.3. Modify Files in /etc

There are several files, mainly in **/etc**, which reflect your system specifics, e.g. how many terminals you have, what speeds they run on, what kinds of terminals are attached and so on.

You should inspect the following files and modify them if necessary: For more information see section 3.3.6, where the files are described in more detail, or manual sections 5 and 8.

Section 2.3

Modify Files in /etc

File	Reason
/etc/bcheckrc	possibility to adapt time to local time zone
/etc/checklist	used by fsck to check your disks
/etc/group	add your own groups, do not delete existing entries
/etc/passwd	add your own users, do not delete existing entries
/etc/inittab	add your terminals and types change the console entry "none"
/etc/issue	will be printed before login
/etc/profile	for definition of TZ
/etc/rc	you may insert # in front of fcookies add your own mount commands, remove # in front of desired utilities
/usr/lib/crontab	used by clock daemon for regular rerun of tasks

You should examine and modify the following files, if you intend using them:

File	Reason
/etc/checkall	fsck for your disks
/etc/gettydefs	for different terminal settings e.g. modem control at the console
/etc/motd	message of the day

The file **/etc/inittab**, specifying on which terminal a *login* should be possible, it's type and default settings for it's line parameters, contains a lot of line blocks prepared for the different hardware configurations. Search for your configuration and check the terminal entries.

E.g. you have two QUME terminals connected to line 1 and 2 of the MUXKE2 board and want to allow a login on the terminals, the entries in **/etc/inittab** should look as follows:

```
# CADMUS 9200 and CADMUS 9520
# 8 serial interfaces on muxke2 and no modem cable
# make MUXKE2
# tty10 is the console (co:)
t11:2:respawn: /etc/getty -h tty11 9600 qume
t12:2:respawn: /etc/getty -h tty12 9600 qume
#####t13:2:respawn: /etc/getty -h tty13 9600 none
#####t14:2:respawn: /etc/getty tty14 9600 none
#####t15:2:respawn: /etc/getty tty15 9600 none
#####t16:2:respawn: /etc/getty tty16 9600 none
#####t17:2:respawn: /etc/getty tty17 9600 none
```

Change the console entry in **/etc/inittab**. As delivered by PCS, the terminal type is set to **none**, in order not to confuse **/etc/getty** when there is an unknown terminal type used as console in Multi-User mode. You should replace **none** by the appropriate terminal type (e.g. **pcsdsg100**, for this specific emulation of a vt100). See **getty(8)** and **term(7)** for more information. The correct terminal type is required for screen oriented programs such as **med** or **vi**.

Make sure that you can login at all terminals. If you have put the right TERM variable in **/etc/inittab**, then each terminal should clear the screen before the login message appears.

If all is set up properly, you may now bring up the system with the kernel of your choice. (See next section 2.4 Virtual and Nonvirtual Systems.)

2.4. Virtual and Nonvirtual Systems

An order for a virtual system contains kernels for the virtual and the nonvirtual system: **vunix** and **nunix**. This section explains how to switch between the two systems.

Several programs exist in two variations - for virtual and nonvirtual systems. These must be copied to there correct places whenever, and only, you change between the two systems.

This is done best by the shell procedure **kernel.run** in the directory **/usr/sys**, see **kernel(8)** for further information. Before you call **kernel.run** you should go in single user mode. To do this go to the root directory (**cd /**) and call **shutdown.sh**.

The following example shows the installation of a virtual kernel with the procedure **/usr/sys/kernel.run**.

kernel.run	Your input
	cd / shutdown.sh
	cd /usr/sys kernel.run
TESTING of a new kernel:	
If you have a WORKING /unix save it!	
Do you want to save your working /unix in /ounix (y,n)?	y
Are you in Single User Mode on the console (y,n)?	y
The following installs the system of your choice	
Do you want to RUN the nonvirtual kernel, nunix (y,n)?	n
Do you want to RUN the virtual kernel, vunix (y,n)?	y
Autoboot /unix with the INIT key.	

Kernel.run first copies the selected kernel (vunix or nunix) to /unix and then copies the programs related to this kernel to their correct places.

3. Getting Started with an Unconfigured System

In this chapter we explain how to load MUNIX V.2/06 into a virgin CADMUS system. The setting up notes pertain to various CADMUS models which differ mainly in disk and tape devices used. The distribution media are magnetic tape or streamer cartridge. The following table lists the models and their devices:

CADMUS model	boot disk	distribution tape	comment
9230	HK,hk	ST,st	4 track streamer
9230	HK,hk	TS,mt	mgtape with DQ 132 controller
9230	HK,hk	TM,mt	mgtape with DQ 130 controller
9240	HK,hk	ST,st	bigger 9230
9520	XT,xt	WT,wt	9 track streamer
9212 Ext	XT,xt	ST,st	2nd disk is RL

We suggest you use the following sequence for the setting-up process:

- 1) Save old software only if you have already used MUNIX Version 1.4 or 1.5. (See section 3.1 Saving Old Software.)
- 2) Prepare the boot disk and load the root filesystem with the help of a special MUNIX kernel: aunix. (See section 3.2 Coldstart.)
- 3) Bring up the system, check the system configuration and load extra software. (See section 3.3 System Installation.)
- 4) Make new MUNIX kernels, specially adapted to your system, if necessary. (See chapter 4 New Kernel.)

3.1. Saving Old Software

This section is only relevant if you have already used a MUNIX operating system. Here we describe which software you should save if you had used either MUNIX 1.4 or MUNIX 1.5.

Since we deliver MUNIX V.2, FORTRAN-77 is no more part of the standard distribution but an optional package. Therefore you must save it, if you still intend using your older version. FORTRAN version 3.1 or newer may be ordered. Your FORTRAN package runs under MUNIX V.2 as well as under MUNIX 1.4 and MUNIX 1.5.

3.1.1. MUNIX 1.4

Save your own files on a backup medium (use cpio). When you read in the files on the new system cpio takes care of the 1024 byte filesystem. In addition, save several system files. The first set of system files should be saved for later examination. They will not be copied back! The second set is saved and later copied back.

The following files are in the first set, which is not copied back, but must be inspected if you have introduced local additions or changes. We advise you to copy them relative to /, in order to avoid copying them over the new files.

File	Reason
<code>/.profile</code>	
<code>/etc/fstab</code>	your file system names
<code>/etc/group</code>	your groups
<code>/etc/keycap</code>	your keyboard capabilities
<code>/etc/passwd</code>	your user entries
<code>/etc/profile</code>	your general user profile
<code>/etc/rc</code>	your mount commands
<code>/etc/termcap</code>	your terminal capabilities
<code>/etc/ttys</code>	your terminal names
<code>/etc/ttytype</code>	your terminal types and speeds
<code>/usr/lib/crontab</code>	your local additions
<code>/usr/sys/conf.h</code>	your system configuration

The following files must be saved to be copied back later:

File	Reason
<code>/usr/lib/uucp/L-devices</code>	UUCP
<code>/usr/lib/uucp/L.sys</code>	UUCP

If you had other local commands in `/bin` or `/usr/bin`, save them separately.

3.1.2. MUNIX 1.5

If your system is a **CADMUS 9230 or 9240**, you have to save all your own software and filesystems on all 65 MB disks, i.e. all devices with major device numbers 0 and 4. You will be able to write them back without difficulty, as the sizing has progressed to bigger filesystems.

If your system is a **CADMUS 9520**, you should save all your own

software in the root filesystem (/, /dev, /bin, /etc), usr filesystem (/usr/local, /usr/bin ...) and usr/lib filesystem.

If your system is a **CADMUS 9212 Ext.**, you should save all your own software in the root filesystem (/, /dev, /bin, /etc), usr filesystem (/usr/local, /usr/bin ...) and usr/lib filesystem. Note that the new operating system MUNIX V.2 will reside on the XT disk, but no longer on the RL disk as under MUNIX 1.5 ! The Minitor 2.4xx cannot boot from the RL disk.

The file /etc/rc must be saved to be inspected later. The following files may be saved and later copied back for comparison. We advise you to copy them relative to /, in order to avoid copying them over the new files.

File	Reason
/etc/bcheckrc	System specific
/etc/group	System specific
/etc/inittab	System specific
/etc/issue	System specific
/etc/passwd	System specific
/etc/profile	System specific
/usr/lib/crontab	System specific
/usr/lib/uucp/L-devices	UUCP
/usr/lib/uucp/L.sys	UUCP

3.2. Coldstart

Coldstart is the term used for bringing up a virgin system, or bringing up a CADMUS system which has had a severe data loss and cannot be brought up by any other means. The Coldstart consists of following steps:

- 1) Boot a special MUNIX kernel (aunix).
- 2) Format your hard disks if necessary and check them for bad blocks.
- 3) Make a file system on the root partition of the boot disk.
- 4) Restore the root file system from the distribution medium.

For the Coldstart procedure you should have the following items handy:

- + The first tape or streamer cartridge of the installation package.
- + The System Log Book of your system which contains
 - the bad block list of your hard disk which is necessary to check or enter the bad blocks. (CADMUS 9520 and 9212 Ext systems only)
 - the PCS supplied configuration sheets of your system:
KONFIGURATOR FUER LOGIKRAHMEN,
HARDWARE CHECKLISTE and
MUNIX CHECKLISTE FUER SYSTEMGENERIERUNG.

You should be familiar with your hardware configuration - especially with the short names of the disk, tape and streamer devices. If your system has one of the devices TM (Magtape with Dilog130 controller), RL, HL (20 MB disk with Andromeda controller) or RX (floppy with Andromeda controller), you should know, whether your system has DMA-wiring (DMA-Verdrahtung), and if so which DMA-extension register is connected to which device. All this information is found in these configuration sheets.

The **boot** tape or cartridge contains the following files:

- 0) aunix
- 1) icckernel
- 2) root file system in dump format
- 3) root file system in cpio format

The root file system in cpio format should never be read in entirely. It is delivered in this format only for the convenience of copying back single files to the root file system in case some have been lost or damaged on the disk. In case the root file system is severely damaged, the safest way to rebuild it is as described here.

begin

3.2.1. CADMUS 9230/9240

If you want to boot from tape, set the magtape drive to
ONLINE

System	Your Input	Comment
	Hit INIT key	
<i>Minitor prompts with</i> <i>Minitor 2.3xx</i>		
<i>Ethernetaddr 08002700xxxx</i>		
<i>Serialnum xxxx</i>		these two messages only for Minitor 2.306
<i>autoboot (hit any key to break)</i>	hit any key	
<i>?bad command</i>		
	ru,	select tape (DQ132 controller)
	rt or	select tape (DQ130 controller)
	→rs	CADMUS 9230/9240 streamer
	/	reads in first file (aunix)
	g	starts program
... ...		system information
		several questions about your hardware configuration follow answer them with
<i>Do you ... for DMA (y,n) ?</i>	y or n	
<i>only 22 bit ... on DER 0 ? (y,n)</i>	n	in general
<i>cold start?</i>	y	in general
<i>aunix menu</i>	y	

3.2.1.1. Check

Choose the function **check** from the aunix menu:

aunix	Your Input	Comment
<i>check = c</i>		
<i>mkfs = m</i>		
<i>restor = r</i>		
<i>xd = x</i>		
<i>quit = q</i>		
<i>select function: check</i>		Disk format and check program

Check formats the disks and checks them for bad blocks. If bad blocks

are found, they are recorded in a bad sector table, and replacement blocks are allocated.

The check program displays a set of devices which it supports and asks for a device name. You enter **hk** and the **disk number** in brackets. The disk number is the physical disk number, as interpreted by the controller.

You have an HK 80 MB disk, which is "seen" by the controller as three physical disks, 2 RK07's and 1 RK06. So enter **hk(0) -w**, **hk(1) -w**, and **hk(2) -w**. For a second disk on the same controller the numbering continues with **hk(3)**, **hk(4)** and **hk(5)**. (The disk number, or physical unit number, must not be confused with major or minor device numbers, or with device names.)

Please note the option **-w** for read-write mode. If check is executed without options then read-only mode is assumed.

After the selection of a device, a menu of functions if displayed and **check** asks for a command. Enter

- f** to format the disk,
- b** to make a bad sector scan, and
- q** to terminate (quit) this section.

Repeat this sequence for all disk numbers, and then enter "exit" to exit from the check program. This will bring you back to the coldstart menu.

3.2.2. CADMUS 9520

System	Your Input	Comment
	Switch key from position ON to position BOOT, release key	
<i>Minitor prompts with Minitor 2.4xx autoboot (hit any key to break)</i>	hit any key	
?bad command		
.	ra	CADMUS 9520 streamer
.	/0	reads in 1st file (aunix)
.	g	starts program
....		system information
		several questions about your hardware configuration follow answer them with
	y or n	
<i>Do you ... for DMA (y,n) ?</i>	n	in general
<i>only 22 bit ... on DER 0 ? (y,n)</i>	y	in general
<i>cold start?</i>	y	
<i>aunix menu</i>		

3.2.2.1. Check

Choose the function **check** from the a unix menu:

aunix	Your Input	Comment
check = c		
mkfs = m		
restor = r		
xd = x		
quit = q		
<i>select function: check</i>		Disk format and check program

Check formats the disks and checks them for bad blocks. If bad blocks are found, they are recorded in a bad sector table on block 0, together with a replacement block. Thus, the disk appears 100% ok. This is a somewhat difficult program to handle. The "gotos" and a default

answer feature should help the user find his way around. The defaults are selected by simply entering an empty line, i.e. hitting <RETURN>.

Some disks are delivered with bad blocks by the manufacturer. For these disks the manufacturer also supplies a "Media Defects List", which records these bad blocks. This "Media Defects List" is appended to your System Log Book by PCS.

Please have this list in front of you when checking the disk.

First, have a look at the bad block list as recorded on the disk itself. The check program prompts you for the **device name** and the **disk number** in brackets. Normally you would have only one drive, which is selected by entering **xt(0) -w**. An optional second drive would have the indication **xt(1)**. Please note the option **-w** for read-write mode. If check is executed without options then read only-mode is assumed.

Thereafter the system displays a set of possible disk-drive types. You have to enter the type of your drive **fujitsu-86** or **maxtor-65**. In order to see the bad block list, skip the next few questions with the help of default answers until the system asks you whether you want to inspect the defect info in block 0.

Answer **y** to this question and the system should give you the DEFECT LIST together with the number of known bad blocks and some additional information. Error messages such as "CANNOT READ BLOCK 0" or "ioctl info is at a prior level SORRY" are possible. If you can get the DEFECT LIST from the disk then go to **COMPARE** else go to **IGNORE**.

COMPARE

Compare the DEFECT LIST as shown on the screen with the "Media Defects List" in your System Log Book.

If there are more bad blocks in the "Media Defects List" than in the DEFECT LIST on the screen, go to **BADBLOCK ENTER**, else go to **BADBLOCK SCAN**.

BADBLOCK ENTER

If there are known bad blocks missing in the DEFECT LIST on the disk, you have to enter the bad blocks. First you have to enter the drive identification as before, e.g. **xt(0) -w** and then the drive type. Skip questions as before until the question

do you want to enter defect info of disk, default = n ? (y/n) :

comes up. Do not get confused by a similar question, which offers you the facility to enter bad blocks by their (physical) sector numbers. Answer the question shown above with **y**. The system displays some remarks as to how the data should be entered and the input can be

started after the prompt:

(decimal) Data; empty line when done>:

After each prompt enter one known bad block not yet recorded on the disk in the form:

cylinder head byte-offset 2 <RETURN>

Note the **2** for "number of bits". Use the decimal data columns (on most sheets given in brackets). After having entered all bad blocks, close the DEFECT LIST by entering an empty line.

The program shows you the updated DEFECT LIST and reformats the disk, finding a replacement block for each bad one.

The formatting is done when the following message appears:

Formatting complete

Go to BADBLOCK SCAN.

IGNORE

If you wish to ignore the error messages and build a new DEFECT LIST, format the disk by entering the drive identification e.g. **xt(0) -w** and the drive type as before. Skip the following questions by entering empty lines until

shall defect info in block 0 be ignored, default = n ? (y/n)

comes up; answer it with **y**. Then skip further questions until the system asks

do you want to enter defect info of disk, default = n ? (y/n) :

Do not get confused by a similar question, which offers you the facility to enter bad blocks by their (physical) sector number. Answer the question shown above with **y**. The system displays some remarks as to how the data should be entered and the input can be started after the prompt:

(decimal) Data; empty line when done>:

After each prompt enter one known bad block not yet recorded on the disk in the form:

cylinder head byte-offset 2 <RETURN>

Note the **2** for "number of bits". Use the decimal data columns (on most sheets given in brackets). After having entered all bad blocks, close the DEFECT LIST by entering an empty line.

The program shows you the updated DEFECT LIST and reformats the disk, taking care to find a replacement block for each bad one.
The formatting is done when the following message appears:

Formatting complete

Go to BADBLOCK SCAN

BADBLOCK SCAN

Like before enter the drive identification, e.g. **xt(0) -w** and the drive type. Skip all following questions until the question

do you want to run a bad block scan ? (y/n)

comes up; answer it with **y**. This feature will take about 30 minutes. The disk will be reformatted if bad blocks are found. Repeat the bad block scan until no more bad blocks are reported.

The last command you enter is:

exit

This brings you back to the coldstart menu.

3.2.3. CADMUS 9212 Ext

System	Your Input	Comment
	Hit INIT key	
<i>Minitor prompts with Minitor 2.4xx autoboot (hit any key to break)</i>	hit any key	
?bad command		
	rs	CADMUS 9200 streamer
	/0	reads in 1st file (aunix)
	g	starts program
.... .		system information
		several questions about your hardware configuration follow answer them with
	y or n	
<i>Do you ... for DMA (y,n) ? only 22 bit ... on DER 0 ? (y,n) cold start? aunix menu</i>	y	in general
	y	in general
	y	

3.2.3.1. Check

Choose the function **check** from the a unix menu:

a unix	Your Input	Comment
check = c		
mkfs = m		
restor = r		
xd = x		
quit = q		
<i>select function: check</i>		Disk format and check program

The check program formats the disks and checks them for bad blocks. If bad blocks are found, they are recorded in a bad sector table on block 0, together with a replacement block. Thus, the disk appears 100% ok. This is a somewhat difficult program to handle. The "gotos" and a default answer feature should help the user find his way around. The defaults are selected by simply entering an empty line, i.e. hitting <RETURN>.

Some disks are delivered with bad blocks by the manufacturer. For

these disks the manufacturer also supplies a "Media Defects List", which records these bad blocks. This "Media Defects List" is appended to your System Log Book by PCS.

Please have this list in front of you when checking the disk.

Firstly, look at the bad block list as recorded on the disk itself. The check program prompts you for the **device name** and the **disk number** in brackets. Normally you would have only one drive, which is selected by entering **xt(0) -w**. An optional second drive would have the indication **xt(1)**. Please note the option **-w** for read-write mode. If check is executed without options then read-only mode is assumed.

Thereafter the system displays a set of possible disk-drive types. You have to enter the type of your drive: **fujitsu-86**. In order to see the bad block list, skip the next few questions with the help of default answers until the system asks you whether you want to inspect the defect info in block 0.

Answer **y** to this question and the system will give you the DEFECT LIST together with the number of known bad blocks and some additional information. Error messages such as "CANNOT READ BLOCK 0" or "ioctl info is at a prior level SORRY" are possible. If you can get the DEFECT LIST from the disk then go to **COMPARE** else go to **IGNORE**.

COMPARE

Compare the DEFECT LIST as shown on the screen with the "Media Defects List" in your System Log Book.

If there are more bad blocks in the "Media Defects List" than in the DEFECT LIST on the screen, go to **BADBLOCK ENTER**, else go to **BADBLOCK SCAN**.

BADBLOCK ENTER

If there are known bad blocks missing in the DEFECT LIST on the disk, you have to enter the bad blocks. First you have to enter the drive identification as before, e.g. **xt(0) -w** and then the drive type. Skip questions as before until the question

do you want to enter defect info of disk, default = n ? (y/n) :

comes up. Do not get confused by a similar question, which offers you the facility to enter bad blocks by their (physical) sector numbers. Answer the question shown above with **y**. The system displays some remarks as to how the data should be entered and the input can be started after the prompt:

(decimal) Data; empty line when done>:

After each prompt enter one known bad block not yet recorded on the

disk in the form:

cylinder head byte-offset **2 <RETURN>**

Note the **2** for "number of bits". Use the decimal data columns (on most sheets given in brackets). After having entered all bad blocks, close the DEFECT LIST by entering an empty line.

The program shows you the updated DEFECT LIST and reformats the disk, finding a replacement block for each bad one.

The formatting is done when the following message appears:

Formatting complete

Go to BADBLOCK SCAN.

IGNORE

If you wish to ignore the error messages and build a new DEFECT LIST, format the disk by entering the drive identification e.g. **xt(0) -w** and the drive type as before. Skip the following questions by entering empty lines until

shall defect info in block 0 be ignored, default = n ? (y/n)

comes up; answer it with **y**. Then skip further questions until the system asks

do you want to enter defect info of disk, default = n ? (y/n) :

Do not get confused by a similar question, which offers you the facility to enter bad blocks by their (physical) sector number. Answer the question shown above with **y**. The system displays some remarks as to how the data should be entered and the input can be started after the prompt:

(decimal) Data; empty line when done>:

After each prompt enter one known bad block not yet recorded on the disk in the form:

cylinder head byte-offset **2 <RETURN>**

Note the **2** for "number of bits". Use the decimal data columns (on most sheets given in brackets). After having entered all bad blocks, close the DEFECT LIST by entering an empty line.

The program shows you the updated DEFECT LIST and reformats the disk, taking care to find a replacement block for each bad one.

The formatting is done when the following message appears:

Formatting complete

Go to BADBLOCK SCAN

BADBLOCK SCAN

Like before enter the drive identification, e.g. **xt(0) -w** and the drive type. Skip all following questions until the question

do you want to run a bad block scan ? (y/n)

comes up; answer it with **y**. This feature will take about 30 minutes. The disk will be reformatted if bad blocks are found. Repeat the bad block scan until no more bad blocks are reported.

The last command you enter is:

exit

This brings you back to the aunix menu.

3.2.4. Mkfs

After checking the disks select the function **mkfs** from the aunix menu.

aunix	Your Input	Comment
<i>check = c</i>		
<i>mkfs = m</i>		
<i>restor = r</i>		
<i>xd = x</i>		
<i>quit = q</i>		
<i>select function:</i>	mkfs	Create a filesystem

With the following answers you create the root filesystem:

mkfs	Your Input	Comment
<i>file system size(counted in 512 Byte Blocks):</i>	16368 or 16384	for HK ← for XT
<i>file system?</i>	hk(0) or xt(0)	for HK ← for XT

Mkfs creates the root filesystem and then returns you to the Coldstart menu. The number in brackets after the disk identifier is the **minor device number**.

3.2.5. Restor

With the **restor** program you can read the root file system onto disk. Note that the disk must already contain a new filesystem made by **mkfs**.

aunix	Your Input	Comment
<i>check = c</i> <i>mkfs = m</i> <i>restor = r</i> <i>xd = x</i> <i>quit = q</i>		
<i>select function:</i>	restor	Restor the root filesystem
<i>restor prompts "Tape?"</i>	ts(0,2), tm(0,2), → st(0,2) or wt(0,2)	DQ132 controller DQ130 controller 9200 series streamer 9520 streamer
<i>restor prompts "Disk?"</i>	→ hk(0), rm(0) or xt(0)	for HK for RM for XT
<i>restor prompts</i> <i>"last chance before</i> <i>scribbling on disk"</i>	Hit <Return>	a few seconds will pass before the tape or streamer cartridge moves
	wait until restor prompts with "End of tape".	

The tape or streamer cartridge rewinds and you will find yourself back in the aunix menu.

3.3. System Installation

In this section we will be assuming that you have a root filesystem on your system, but that no further software has been loaded. Here we describe how to bring up such a system, how to check the configuration, create further filesystems, and read in more software.

3.3.1. Startup MUNIX

After you read in the tape you can go on to startup MUNIX by leaving the function menu.

aunix	Your Input	Comment
check = c		
mkfs = m		
restor = r		
xd = x		
quit = q		
select function:	quit	Leave the a unix menu
want to startup UNIX?	y	autoboot a unix

3.3.1.1. Root and Swap Devices

aunix will ask you for the root and swap disk, the starting block and the size of the swap space. The following sections contain the necessary information for the various CADMUS models.

→ **CADMUS 9230/9240:**

Starting with MUNIX V.2 we introduce 3 classes of swap space. These standard settings are shown below:

Class 1:

For "small" systems running only nonvirtual MUNIX, less than 6 users and less than 1.5 MB of main memory a small swap space of about 5 MB may be configured (SMALLHK). The root filesystem is on a HK disk with minor device number 0, and the swap space resides on the HK disk with minor device number 18, starts at block 0 and has the size of 10560 Blocks.

Section 3.3

System Installation

ROOT	type of root	unit of root	type of swap	unit of swap	start of swap	size of swap
HK	4	0	4	18	0	10560

Class 2:

For "medium" systems running virtual MUNIX and 1 or 2 MB of main memory a medium swap space of about 8 MB may be configured on a HK only (MEDIUMHK). The root system is on a HK disk with minor device number 0, and the swap space resides on the HK disk with minor device number 16, starts at block 0 and has the size of 16368 Blocks.



ROOT	type of root	unit of root	type of swap	unit of swap	start of swap	size of swap
HK	4	0	4	16	0	16368

Class 3:

For "big" systems running virtual MUNIX either with more than 8 users, or 3-4 MB of main memory, or AI, or CAD applications, or 2 and more bitmap terminals a big swap space of about 13 MB may be configured (BIGHK). For a configuration with a big size swap space (MEDIUMHK) The root system is on a HK disk with minor device number 0, and the swap space resides on the HK disk with minor device number 22, starts at block 0 and has the size of 26928 Blocks.

ROOT	type of root	unit of root	type of swap	unit of swap	start of swap	size of swap
HK	4	0	4	22	0	26928

CADMUS 9520 and CADMUS 9212 Ext:

For a standard configuration on these models the root is on the XT disk with minor device number 0, the swap area is on the XT disk with minor device number 1, the swap area starts at block 0 and is 16384 blocks in size.

ROOT	type of root	unit of root	type of swap	unit of swap	start of swap	size of swap
XT	75	0	75	1	0	16384

Other configurations may be possible (see "New Kernel").

When the kernel asks you whether it should go into multi user mode, answer **n** for no.

System	Your Input
	<i>going Multi-user mode ? (y/n): n</i>
	<i>to go multi-user type "init 2".</i>

file system check

For safety reasons you should do a file system check of the root filesystem:

System	Your Input
CADMUS 9230/9240	fsck -y /dev/hk0.0
CADMUS 9520	fsck -y /dev/xt0a
CADMUS 9212 Ext	fsck -y /dev/xt0a

If there are any inconsistencies they will be corrected automatically and after the message BOOT UNIX NO SYNC, the system will return to the monitor. Boot **/aunix** once again.

If the root filesystem you have read in is not suitably configured for your system, you may, quite likely, get an error message such as **cannot open /dev/xxx**. In this case you should set the correct date

(section 3.3.2), create the particular special file (section 3.3.3) and enter the above command **fsck ...** once more.

3.3.2. Setting the Correct Date

After having read in the root filesystem, the date will be wrong. Quite likely it will be some day in 1970. Setting the date is absolutely necessary. Set the correct date with the command

date mmddhhmmyy

mm is the month number
dd is the day number in the month
hh is the hour number (24 hour system)
mm is the minute number
yy is the last two digits of the year

For example: **date 0415093086** sets the date to Apr 15, 1986, 9:30 am.

3.3.3. Special Files

Look at the entries in directory **/dev**.

ll /dev

The special files in this directory should reflect your hardware configuration. If your devices are configured correctly, you can continue to the next section, concerning filesystems and disk-partitioning.

If your devices are not configured correctly, you have to call **make** for the needed entries. For this purpose, use the provided driver checklist for system generation (MUNIX CHECKLISTE FUER SYSTEMGENERIERUNG).

First, you have to go to **/dev**.

cd /dev

For each marked entry in the checklist type:

make XX

where **XX** is something like **MUXKE2, ST, HK, XT, WT, ...**

3.3.3.1. Disk Partitioning and Swap Device

CADMUS 9230/9240:

The following commands partition the disk (dev/hk??.?, dev/swap) for the three classes of small, medium and big swap systems (see Releasenotes). Devices are made for the usr disk on the 1st RK07, for your own filesystem on the 2nd RK07 and for an additional filesystem in case of small, or medium size swap space. This makefile also makes devices for a 2nd Winchester disk, which will do no harm even if you have just 1 Winchester disk.

Device	Commands	Comment
HK	make SMALLHK	disk partitioning for a small
HK	make MEDIUMHK	medium
HK	make BIGHK	big size swap space

CADMUS 9520 and CADMUS 9212 Ext:

There is only the standard way of partitioning the disk. The following command makes devices for the usr and usr/lio disk, the swap space and your own filesystem.

Device	Command	Comment
XT	make XT	standard disk partitioning

3.3.4. Filesystems

Next you must **make additional filesystems** on the disk(s) and **mount** them on their respective mount directories. We describe the standard disk-partitioning as recommended by us.

3.3.4.1. CADMUS 9230/9240

First of all make the filesystem for **/usr**. The following commands make a filesystem on a device with the name hk0.1, do a file consistency check on the new filesystem and mount it on the **usr** directory:

Device	Command	Comment
HK	<code>mkfs /dev/rhk0.1 37224 4 400</code> <code>fsck /dev/rhk0.1</code> <code>mount /dev/rhk0.1 /usr</code>	filesystem for /usr in all cases

Other additional filesystems are intended for your deliberate use.

Organization of Your Filesystems:

Earlier we suggested you use the first RK07 for PCS provided software, i.e. the root and usr filesystems. The RK06 is used partly (small and medium swap space) or completely for swap space. That leaves at least the entire second RK07 for your use.

The manual page *hk(4)* gives a list of all possibilities for HK disk partitioning. The following sections describe two of these possibilities and also consider other aspects, such as backup and device names.

1) One Filesystem on a RK07

This filesystem has a size of 53636 blocks, i.e. about 26 MB. Type

→ `mkfs /dev/rhk1.7 53636 4 400`

The device **hk1.7** has already been made (see section 3.3.3. Special Files). A file system of this size may present some problems when backing up on 15-18 MB streamer cartridges. In this case you may either backup only some directories at a time or use a magtape as backup medium.

2) Two Filesystems on a RK07

These two filesystems are roughly of the same size of 13 MB each. You first have to **create** the special files and then **make** the filesystems.

```
/etc/mknod /dev/hk1.6 b 4 14
/etc/mknod /dev/rhk1.6 c 11 14
/etc/mknod /dev/hk1.3 b 4 11
/etc/mknod /dev/rhk1.3 c 11 11
```

```
mkfs /dev/rhk1.6 26928 4 400
mkfs /dev/rhk1.3 26664 4 400
```

Other possibilities follow the same scheme. We also suggest, not to just type in the mknod commands for your selection of disk partitioning but to have a shell file or a make file in /dev.

If you use a small or medium size swap space you still have room for an **additional file system on the RK06**. The respective device has

already been made.

SMALLHK

mkfs /dev/rhk2.0 16368 4 400

MEDIUMHK

→ **mkfs /dev/rhk2.2 10560 4 400**

Example:

A a possible disk partitioning for a CADMUS 9230 system with a medium size swap space.

major	4	4			
minor	0	1			
blocks	16384	37224			
what					
dev	root hk0.0	usr hk0.1	first RK07		

major	4	4			
minor	15	15			
blocks	53636				
what					
dev	free hk1.7		second RK07		

major	4	4			
minor	16	18			
blocks	16384	10560			
what					
dev	swap swap	free hk2.2	RK06		

~~3.3.4.2.~~ CADMUS 9520 and CADMUS 9212 Ext

The following commands make filesystems for `/usr` and `/usr/lib`, do a file system consistency check and mount them on their respective mount directories:

Device	Commands	Comment
XT:	<code>mkfs /dev/xt0c</code> <code>fsck /dev/xt0c</code> <code>mount /dev/xt0c /usr</code>	<code>32768 6 400</code> filesystem for <code>/usr</code>
	<code>mkfs /dev/xt0e</code> <code>fsck /dev/xt0e</code> <code>mkdir /usr/lib</code> <code>mount /dev/xt0e /usr/lib</code>	<code>32768 6 400</code> filesystem for <code>/usr/lib</code>

The other additional filesystem is intended for your deliberate use and is created and checked by the commands:

```
mkfs/dev/xt0f 17114 6 400
fsck /dev/xt0f
```

See `xt(4)` for another way of partitioning the disk.

If your system is a CADMUS 9212 Ext, you have now additional filesystems on the RL disk. See `rl(4)` for your way of partitioning the disk.

3.3.5. Reading Filesets

The standard distribution of MUNIX V.2 has been grouped into 16 filesets, one for the root filesystem (`rootlist`) and 15 for the `usr` filesystem. This allows the system administrator to tailor the system to the local requirements, i.e. space or usage constraints.

In the directory `/usr/local/filesets` you will find several files ending with '`list`' (e.g. `kernel_vlist`). These files contain the pathnames of the files distributed with your system.

If you want to use files from other filesets (like `man`, `lex`, or `uucp`), you have to read them in from your distribution media.

The order of the filesets on the cartridge is as follows:

list	size in blocks	size in MB	contents
kernel_v	6858	3.42	all kernel related files for virtual systems
or kernel_s	4333	2.16	all kernel related files for non-virtual systems
usr	4860	2.43	most usr files
terminfo	1756	0.87	screen handling
icc	384	0.19	icc related software
acct	697	0.34	Accounting
 games	1813	0.90	Games X
graphic	376	0.18	simple graphics
lexyacc	985	0.49	lex, yacc, awk
lp	766	0.38	printer spooler
man	3788	1.89	manual pages
nroff	764	0.38	simple formatting
testmon	890	0.44	testmonitor
tmac	1183	0.59	necessary macros for nroff and troff
ucb	1905	0.95	Berkeley utilities
uucp	809	0.40	UNIX-UNIX Communication

When reading in any fileset, note the following:

- you must be superuser
- login root**
- don't use the C-Shell but the Bourne Shell (standard shell).
- go to **/** in all cases:
- cd /**
- each fileset is one distinct **cpio** file
- if reading in filesets from streamer cartridge, first rewind the streamer cartridge and then skip the filesets you do not need.

Device	Commands	Comment
CADMUS 9230/9240 and CADMUS 9212 Ext:		
tape:	/etc/mt rew /etc/mt fsf <i>n</i> cpio -iBmvd < /dev/nrmt0	rewind tape skip <i>n</i> filesets on tape read a fileset without rewinding
streamer:	</dev/rst0 stskip <i>n</i> cpio -iSmvd < /dev/nrst0	rewind streamer cartridge skip <i>n</i> filesets on cartridge read a fileset without rewinding
CADMUS 9520:		
streamer	< /dev/rwt0 wtskip <i>n</i> cpio -iSmvd < /dev/nrwt0	rewind streamer cartridge skip <i>n</i> filesets on cartridge read a fileset without rewinding

Continue by alternating the skip and read commands according to your needs.

You may want to write a small shell loop which reads in a number of filesets on a CADMUS 9200:

```
for i in 0 1 2 3 4 5 6 7 8 9 10
do
cpio -ivmdS </dev/nrst0
done
```

The above commands read 11 filesets from a streamer cartridge.

Important:

It is recommended that at least the first 3 filesets exist on your system. The first fileset, named kernel_v or kernel_s, contains all the files which are necessary to make a new kernel, or to switch to the virtual system. The second fileset, usr, contains most files in /usr/bin. It is also recommended to read in the terminfo fileset as it contains the description of terminal characteristics. If you want to use additional utilities like lex + awk or uucp, you have to read in the corresponding filesets from tape or cartridge.

If you wish to check, save, or remove filesets, we suggest you use the shellscripts provided in /usr/local/filesets and the lists /usr/local/filesets/*list.

The entries in this lists have been changed to allow checks for correctness. Each pathname is followed by 2 numbers, the output of the program sum. The shell script ckft may thus check for existence and consistency of all files belonging to a fileset.

This was done for all filesets belonging to the MUNIX standard

distribution, we will do the same for all optional packages in the future.

ckft <fileset> ...

Check a fileset for completeness and consistency on disk.

cpioft <unit> <fileset> ...

Copy a fileset to unit, where unit may be a streamer (st, wt or is), a magtape (mt), or a directory (directory name).

rmft <fileset>

Remove a fileset from disk.

Another shell script /usr/local/copyboot puts the 2 startup files and a dump of the root filesystem on a streamer or magtape. This shell script may be used for a backup of your root filesystem after you have added all your local additions.

3.3.6. Adapting /etc Files

There are several files, mainly in /etc, which reflect your system specifics, e.g. how many terminals you have, what speeds they run on, what kinds of terminals are attached and so on. For a description of the files, see **manual sections 5 and 8**.

You should inspect the following files and modify them if necessary (If you used MUNIX 1.4 or 1.5, you should compare these with your backups (see section 3.1):

```
/etc/bcheckrc  
/etc/checklist  
/etc/group  
/etc/passwd  
/etc/inittab  
/etc/issue  
/etc/profile  
/etc/rc  
/usr/lib/crontab  
/etc/checkall  
/etc/gettydefs  
/etc/motd
```

The following list describes these files briefly:

/etc/bcheckrc

contains commands which are executed by the system before the filesystems are mounted. You may have to change the definition of TZ to comply with your local time zone.

/etc/checklist

contains a list of the disks on which a filecheck is made before mounting. Entries are one to a line and the file should contain no blank lines.

Example:

Entries for a CADMUS 9230 system (with medium swap space and one filesystem on the second RK07):

```
/dev/hk0.0  
/dev/rhk0.1  
/dev/rhk1.7  
/dev/rhk2.2
```

/etc/group

contains entries of user groups. Enter your local groups but do not change the existing entries. For a more detailed description see manual section 5

/etc/passwd

contains a list of the system users. Add your local users but do not delete the existing entries. Never delete the root entry! A user with password **xxxx** has no system access. You may delete **xxxx** or use the command "**passwd user**" to give him a new password. For a more detailed description see manual section 5.

/etc/inittab

The file **/etc/inittab**, specifying on which terminal a **login** should be possible, it's type and default settings for it's line parameters, contains a lot of line blocks prepared for the different hardware configurations. Search for your configuration and delete the comment characters ##### at the beginning of the needed terminal entries.

Change the console entry. As delivered by PCS, the terminal type is set to **none**, in order not to confuse **/etc/getty** when there is an unknown terminal type used as console in Multi-user mode. **None** should be replaced by the appropriate terminal type (e.g. **pcs-dsg100** for this specific emulation of a vt100). See **getty (8)** and **term (7)** for more information. The correct terminal type is required for screen-oriented utilities such as **med** or **vi**. If the correct **TERM** variable in **/etc/inittab** has been set for all terminals each terminal should clear the screen before the login message is displayed.

Example: Supposed you have two QUME terminals connected to line 1 and 2 of the MUXKE2 board and want to allow a login on the terminals, search for the following lines in /etc/inittab and change them as shown below: (the line "# make MUXKE2" refers to the corresponding device entries in /dev/makefile, see section 3.3.3 Special Files.)

Entries with comment characters:

```
# CADMUS 9200 and CADMUS 9520
# 8 serial interfaces on muxke2 and no modem cable
# make MUXKE2
# tty10 is the console (co:)
#####t11:2:respawn: /etc/getty -h tty11 9600 none
#####t12:2:respawn: /etc/getty -h tty12 9600 none
#####t13:2:respawn: /etc/getty -h tty13 9600 none
#####t14:2:respawn: /etc/getty tty14 9600 none
#####t15:2:respawn: /etc/getty tty15 9600 none
#####t16:2:respawn: /etc/getty tty16 9600 none
#####t17:2:respawn: /etc/getty tty17 9600 none
```

Changed entries:

```
# CADMUS 9200 and CADMUS 9520
# 8 serial interfaces on muxke2 and no modem cable
# make MUXKE2
# tty10 is the console (co:)
t11:2:respawn: /etc/getty -h tty11 9600 qume
t12:2:respawn: /etc/getty -h tty12 9600 qume
#####t13:2:respawn: /etc/getty -h tty13 9600 none
#####t14:2:respawn: /etc/getty tty14 9600 none
#####t15:2:respawn: /etc/getty tty15 9600 none
#####t16:2:respawn: /etc/getty tty16 9600 none
#####t17:2:respawn: /etc/getty tty17 9600 none
```

If you intend running the console with modem control you should change the console entry in /etc/gettydefs (see below).

/etc/issue

This is the message which appears before the login prompt and may be changed as desired.

/etc/profile

You may have to change the definition of TZ to comply with your local time zone.

If you do not intend running fcookies after login, enter a # at the beginning of the following lines in the file:

```
if test -x /usr/ucb/fcookie
then /usr/ucb/fcookie
fi
```

/etc/rc Enter your mount commands under the line

: put mounts here

eg. for a CADMUS 9230 system (with medium swap space)

/etc/mount /dev/hk0.1 /usr
/etc/mount /dev/hk2.2 /usr/lib

Remove the # in front of the utilities you need, for example before the lines for the lp spooler.

/usr/lib/crontab

contains commands which must be rerun regularly or on specific days. This file must be edited to reflect your local requirements. To use the existing entries, remove the comment characters (#). Execute the command **crontab /usr/lib/crontab** to have cron recognize the changes. Note that **cron** must be running. You can start cron in **/etc/rc**.

/etc/checkall

You most likely won't need to edit this file unless you intend using a filecheck other than that described in **/etc/checklist**.

/etc/gettydefs

contains terminal parameters which are set when bringing up the system. You will only need to change this file if you intend using terminals with other than standard parameters, or if you intend running the console with modem control. In the latter case, the parameter **CLOCAL** must be removed from the **console** entry.

console entry as delivered by PCS:

console# B9600 HUPCL PARENB CS7 OPOST ONLCR CLOCAL #
B9600 SANE IXANY TAB3 CLOCAL #Console Login: #console

console entry changed to use modem control:

console# B9600 HUPCL PARENB CS7 OPOST ONLCR #
B9600 SANE IXANY TAB3 #Console Login: #console

/etc/motd

contains the message of the day. **/etc/motd** is displayed every time the file **.profile** is read, eg. after login, or while going in single user mode. **/etc/motd** can be edited to display system changes, messages to users etc.

3.4. Loading Unix

The root filesystem you have read in should contain several unix kernels. **/unix**, which is configured for your system, **/vunix** the virtual kernel configured for your system, **/nunix** the nonvirtual kernel configured for your system, and **/aunix**, which can run on a variety of disks and terminals and be used for the coldstart of a CADMUS system.

An order for a nonvirtual system contains unix, nunix and aunix. An order for a virtual system contains all 4 unix kernels with unix being a copy of nunix.

Section 3.4.1 illustrates the choice of running the virtual or nonvirtual kernel and sections 3.4.2 the different loading processes on the different CADMUS models.

If there is no kernel suitably configured for your system, you must first generate a new kernel, see chapter 4 New Kernel.

3.4.1. Installation of the Virtual or Nonvirtual Kernel

This section explains how to install the virtual or nonvirtual system. An order for a virtual system contains kernels for the virtual and the nonvirtual system: **vunix** and **nunix**.

Several programs exist in two variations - for virtual and nonvirtual systems. These must be copied to their correct places. This is done best by the shell procedure **kernel.run** in the directory **/usr/sys**, see **kernel(8)** for further information.

The following example shows the installation of a virtual kernel with the shell procedure **/usr/sys/kernel.run**.

We assume you are still in single user mode and running a unix as through the installation of the rootfilesystem. Therefore you need not save any kernel.

kernel.run	Your input
	<code>cd /usr/sys kernel.run</code>
<i>TESTING of a new kernel:</i>	
<i>If you have a WORKING /unix save it!</i>	
<i>Do you want to save your working /unix in /ounix (y,n)?</i>	n
<i>Are you in Single User Mode on the console (y,n)? The following installs the system of your choice</i>	y
<i>Do you want to RUN the nonvirtual kernel, nunix (y,n)?</i>	n
<i>Do you want to RUN the virtual kernel, vunix (y,n)?</i>	y
<i>Autoboot /unix with the INIT key.</i>	

Kernel.run first copies the selected kernel (vunix or nunix) to /unix and then copies the programs related to this kernel to their correct places.

As the minitor versions 2.301, 2.400 and higher have autoboot capability, /unix will be loaded automatically after hitting the INIT key, or by turning the key to the BOOT position. The following sections illustrate the different loading processes on the different CADMUS models.

Note that after this first installation the different programs for the virtual and nonvirtual system must be copied to their correct places whenever, and only, you change between the two systems. (See section 4.3 Virtual and Nonvirtual Systems)

3.4.2. Loading Unix on Different CADMUS Systems

The following sections describe the different loading processes on the different CADMUS models.

3.4.2.1. CADMUS 9230/9240

System	Your Input	Comment
	Hit INIT key	
<i>Minitor prompts with Minitor 2.3xx Ethernetaddr 08002700xxxx Serialnum xxxx autoboot (hit any key to break)</i>	nothing	these two messages only for Minitor 2.306 a few seconds will pass before the following message appears autoboot
<i>rk0/unix ... INIT:SINGLE USER going Multi-user mode? (y/n): Is the date ... correct? (y or n)</i>	y	System information Correct dates are entered: mmddhhmmmyy, (month first)
<i>Do you want to check the filesystems? (y or n) ... pcs (this is V.2/06) Console Login:</i>	y	fsck information
	root	

3.4.2.2. CADMUS 9520

System	Your Input	Comment
	Switch key from position ON to position BOOT, release key	
<i>Minitor prompts with Minitor 2.4xx autoboot (hit any key to break)</i>	nothing	a few seconds will pass before the following message appears autoboot System information
<i>rc0/unix ... INIT:SINGLE USER going Multi-user mode? (y/n): Is the date ... correct? (y or n)</i>	y	Correct dates are entered mmddhhmmyy, (month first)
<i>Do you want to check the filesystems? (y or n) ... pcs (this is V.2/06) Console Login:</i>	y root	fsck information

3.4.2.3. CADMUS 9212 Ext

System	Your Input	Comment
<i>Minitor prompts with Minitor 2.4xx autoboot (hit any key to break)</i>	Hit INIT key nothing	a few seconds will pass before the following message appears autoboot
<i>rc0/unix</i>		System information
<i>... INIT:SINGLE USER going Multi-user mode? (y/n): Is the date ... correct? (y or n)</i>	y y	Correct dates are entered: mmddhhmmyy, (month first)
<i>Do you want to check the filesystems? (y or n) ... pcs (this is V.2/06) Console Login:</i>	y root	fsck information

3.4.3. Restoring

If you have saved own software (see section 3.1 Saving Old Software) you can read it in now.

4. New Kernel

This section describes how to make new MUNIX kernels, which are specifically adapted to your hardware. First you should read the manual page *newconf(8)* and try out the shell procedure */etc/newconf* in your home directory or under */tmp*. Then go in the directory */usr/sys* where kernels are made, call */etc/newconf* to make your configuration files and generate new MUNIX kernels. The rest of this section tries to help you answering the questions in *newconf* properly. You may wish to read through the whole section before starting *newconf*.

4.1. /etc/newconf

MUNIX V.2 *newconf* offers increased functionality. This includes more hardware drivers (see Release Notes) and also "drivers" which are not related to hardware but to **software** which you might want to include in your kernel. If you want to keep the kernel as small as possible and do not have any need for the software extensions answer **n** to the questions.

It is advisable to use the PCS supplied "KONFIGURATOR FUER LOGIKRAHMEN", "HARDWARE CHECKLISTE" and "MUNIX CHECKLISTE FUER SYSTEMGENERIERUNG" in conjunction with *newconf*.

4.1.1. Drivers

A first set of questions covers mandatory drivers. Later questions cover functions, each question belonging to one topic. The line "choice of: (y/n):" gives you the chance to skip the entire topic (answer **n**). Answer **y** will present each of the functions singly.

Ethernet drivers are used by the local area networks MUNIX/NET or Newcastle Connection. MUNIX/NET on CADMUS 92xx models generally needs U3COM; on CADMUS 9600 and 9900 models ICC_UNISON is needed. For further information about MUNIX/NET see the pertaining documentation.

The communication packets X.25, SNA and RJE need the respective drivers included.

The following paragraphs list parameters with references to other sections for further information.

MESG	This parameter facilitates the inter task communication of mechanism of "messages", see <i>msgctl(2)</i> , <i>msgget(2)</i> , <i>msgop(2)</i> , <i>intro(2)</i> and <i>ipcs(1)</i> .
SEMA	This parameter facilitates semaphores, see <i>semctl(2)</i> , <i>semget(2)</i> , <i>semop(2)</i> , <i>intro(2)</i> and <i>ipcs(1)</i> .
SHMEM	This parameter facilitates shared memory, see <i>shmctl(2)</i> , <i>shmget(2)</i> , <i>shmat(2)</i> , <i>shmdt(2)</i> , <i>intro(2)</i> and <i>ipcs(1)</i> .
SBP	The Simple Block Port Interface is another communication protocol, see <i>sbp(4)</i>
SXT	This parameter is required for the use of the layered shell, see <i>shl(1)</i> .
RLOCK	This parameter includes the code for the John Bass record locking scheme, mandatory for packet UNIFY, see <i>lockf(2)</i> .
VTTY	This parameter is required for the use of the packet Window Shell, see <i>wsh(1)</i> .

4.1.2. 18 Bit Controllers

There is a set of DMA controllers that may be operated with different address widths. The standard settings for CADMUS 92xx systems is shown below:

Device	DMA address bits
HL	18
RL	18
TM	18
RX	18

Deviations from this table occur very seldom.

4.1.2.1. DMA wiring

Standard CADMUS systems do not have DMA wiring. Only if you have 18 bit controllers the system may have DMA wiring. If your system was wired it should be marked on the KONFIGURATOR. If there is DMA wiring, follow the following rule:

All controllers up to (and including) the first slot marked with an * use DMA extension register 0, the following controllers up to (and

including) the second slot marked with * use DMA extension register 2, the remaining controllers use DMA extension register 3.

4.1.3. Root and Swap

Possible root devices for CADMUS models we describe here are: HK and XT.

4.1.3.1. CADMUS 9230/9240

Starting with MUNIX V.2 we introduce 3 classes of swap space. These standard situations are shown below.

Class 1:

For "small" systems running only nonvirtual MUNIX, and less than 8 users a small swap space of about 5 MB may be configured.

system	input
<i>choose ROOT device:</i>	HK
<i>unit ... of ROOT):</i>	0
<i>choose SWAP device:</i>	HK
<i>unit ... of SWAP):</i>	18
<i>SWAP area starts ... :</i>	0
<i>no. of blocks in SWAP area:</i>	10560

Class 2:

For "medium" systems running virtual MUNIX and 1 to 2 MB of main memory a medium swap space of about 8 MB may be configured.

system	input
<i>choose ROOT device:</i>	HK
<i>unit ... of ROOT):</i>	0
<i>choose SWAP device:</i>	HK
<i>unit ... of SWAP):</i>	16
<i>SWAP area starts ... :</i>	0
<i>no. of blocks in SWAP area:</i>	16368

Class 3:

For "big" systems running virtual MUNIX either with more than 8 users, or 3-4 MB of main memory, or AI, or CAD applications, or 2 and more bitmap terminals a big swap space of about 13 MB may be configured.

system	input
<i>choose ROOT device:</i>	HK
<i>unit ... of ROOT):</i>	0
<i>choose SWAP device:</i>	HK
<i>unit ... of SWAP):</i>	22
<i>SWAP area starts ... :</i>	0
<i>no. of blocks in SWAP area:</i>	26928

4.1.3.2. CADMUS 9520 and CADMUS 9212 Ext.

There is only one size of swap space for these systems.

system	input
<i>choose ROOT device:</i>	XT
<i>unit ... of ROOT):</i>	0
<i>choose SWAP device:</i>	XT
<i>unit ... of SWAP):</i>	1
<i>SWAP area starts ... :</i>	0
<i>no. of blocks in SWAP area:</i>	16384

4.1.4. Questions which are not asked**BMT as console**

If you use a Bit-Map as console and have a muxke, muxke2 or dh follow the instructions 12.5 in *Releasenotes for CADMUS 2200*.

MOT_IEEE

This includes the code for double precision floating point, see *fp(3)*, and is always defined.

4.2. Generate New Kernels

After completion of `/etc/newconf`, you should look at the files `/usr/sys/conf.h`, `/usr/sys/conf.modul` and `/usr/sys/name.c` and edit them if you want to deviate from the standard values supplied there.

In this directory there are two shell procedures, `kernel.gen` and `kernel.run`, which assist you in creating kernels and running virtual or nonvirtual systems. See `kernel(8)` for further information.

The makefile in `/usr/sys` generates two kernels, a virtual and a nonvirtual, `/vunix` and `/nunix`. If you do not have a QU 68050 processor board, you can not use the virtual kernel. After the make, you have one or two new kernels `/nunix` and/or `/vunix`.

4.3. Virtual and Nonvirtual Systems

An order for a virtual system contains kernels for the virtual and the nonvirtual system: `vunix` and `nunix`. This section explains how to switch between the two systems.

Several programs exist in two variations - for virtual and nonvirtual systems. These must be copied to there correct places whenever, and only, you change between the two systems.

This is done best by the shell procedure `kernel.run` in the directory `/usr/sys`. See `kernel(8)` for further information. Before you call `kernel.run` you should go in single user mode. To do this go to the root directory (`cd /`) and call `shutdown.sh`.

The following example shows the installation of a virtual kernel with the procedure `/usr/sys/kernel.run`.

kernel.run	Your input
	cd /
	shutdown.sh
	cd /usr/sys
	kernel.run
<i>TESTING of a new kernel:</i>	
<i>If you have a WORKING /unix save it!</i>	
<i>Do you want to save your working /unix in /ounix (y,n)?</i>	y
<i>Are you in Single User Mode on the console (y,n)?</i>	y
<i>The following installs the system of your choice</i>	
<i>Do you want to RUN the nonvirtual kernel, nunix (y,n)?</i>	n
<i>Do you want to RUN the virtual kernel, vunix (y,n)?</i>	y
<i>Autoboot /unix with the INIT key.</i>	

Kernel.run first copies the chosen kernel (vunix or nunix) to /unix and then copies the programs related to this kernel to their correct places.

5. Installing MUNIX V.2/6a as Update to MUNIX V.2/04

MUNIX V.2/6a is a release and not a version. It may thus be installed as an update over V.2/04 or the preleases V.2/05 and V.2/5a which were only delivered for CADMUS 9600 and CADMUS 9900 systems. Updating requires about an hour.

The following explains the updating sequence to the system administrator. Please login as **root** and go in **single user mode**. The **usr** filesystem and the **usr/lib** filesystem (if it is an extra filesystem) must be mounted.

The following commands must be executed before reading the update magtape or streamer cartridge:

Your Input	Comment
<code>mv /etc/inittab /etc/inittab.my</code>	save programs for later comparison:
<code>mv /etc/gettydefs /etc/gettydefs.my</code>	
<code>mv /etc/rc /etc/rc.my</code>	
<code>mv /.profile /.profile.my</code>	
<code>mv /usr/lib/uucp/L-devices /usr/lib/uucp/L-devices.my</code>	
<code>cp /etc/passwd /etc/passwd.my</code>	do not move,
<code>cp /etc/group /etc/group.my</code>	copy instead
<code>rm /usr/nbin/*</code>	remove some files:
<code>rm /usr/vbin/*</code>	
<code>rm /etc/ioctl.syscon</code>	
<code>rm /usr/bin/uname</code>	might not exist
<code>rm /bin/iwctrl</code>	might not exist
<code>rm /bin/cu</code>	might not exist
<code>mv /bin/cpio /bin/cpio.sav</code>	program is needed for reading tape or streamer

The following commands should be given with great care. If the system crashes before **cp** finishes, you will not be able to reboot.

Your Input
<code>mv /bin/sh /bin/sh.sav ; cp /bin/sh.sav /bin/sh</code>
<code>mv /etc/init /etc/Init ; cp /etc/Init /etc/init</code>

5.1. Reading Tape or Streamer

This update is read like any usual fileset. Make sure you are in /. Note the u option, you will not be able to overwrite /bin/sh and /etc/init otherwise.

Device	Commands	Comment
<u>CADMUS 9230/9240 and CADMUS 9212 Ext:</u>		
streamer:	cd / </dev/rst0 cpio.sav -iSmvdu < /dev/rst0	rewind streamer cartridge read the fileset with rewind
magtape:	cd / /etc/rew cpio.sav -iBmvdu < /dev/rmt0	rewind magtape read the fileset with rewind
<u>CADMUS 9520:</u>		
streamer:	cd / </dev/rwt0 cpio.sav -iSmvdu < /dev/rwt0	rewind streamer cartridge read the fileset with rewind
<u>CADMUS 9600 and 9900:</u>		
streamer:	cd / </dev/ris0 cpio.sav -iSmvdu < /dev/ris0	IS format rewind streamer cartridge read the fileset with rewind
streamer:	isctrl -q. </dev/ris0 cpio.sav -iSmvdu < /dev/ris0 isctrl -i	ST format set streamer to ST format rewind streamer cartridge read the fileset with rewind reset streamer to IS format

5.2. Administrative Files

The files /etc/gettydefs, /etc/inittab, /etc/passwd, /etc/group and /etc/rc are new. Compare them with your .my files. Do not copy the .my version of these files but adapt the distributed files to your local conditions with help of the .my files.

ATTACHMENT 11.2

```
struct sysinfo {
    time_t          cpu[4];
#define CPU_IDLE      0
#define CPU_USER      1
#define CPU_KERNEL    2
#define CPU_WAIT      3
    time_t          wait[3];
#define W_IO          0
#define W_SWAP        1
#define W_PIO         2
    long            bread;
    long            bwrite;
    long            lread;
    long            lwrite;
    long            phread;
    long            phwrite;
    long            swapin;
    long            swapout;
    long            bswapin;
    long            bswapout;
    long            pswitch;
    long            syscall;
    long            sysread;
    long            syswrite;
    long            sysfork;
    long            sysexec;
    long            runque;
    long            runocc;
    long            swpque;
    long            swpocc;
    long            iget;
    long            namei;
    long            dirblk;
    long            readch;
    long            writech;
    long            revint;
    long            xmtint;
    long            mdmint;
    long            rawch;
    long            canch;
    long            outch;
    long            msg;
    long            sema;
};
```

ATTACHMENT 11.3

The derivation of the reported items is given in this attachment. Each item discussed below is the data difference sampled at two distinct times t_2 and t_1 .

CPU Utilization

$$\%-\text{of-cpu-x} = \text{cpu-x} / (\text{cpu-idle} + \text{cpu-user} + \text{cpu-kernel} + \text{cpu-wait}) * 100$$

where cpu-x is cpu-idle, cpu-user, cpu-kernel (cpu-sys), or cpu-wait.

Cached Hit Ratio

$$\%-\text{of-cached-I/O} = (\text{logical-I/O} - \text{block-I/O}) / \text{logical-I/O} * 100$$

where cached I/O is cached read or cached write.

Disk or Tape I/O Activity

$$\begin{aligned}\%-\text{of-busy} &= \text{I/O-active} / (t_2 - t_1) * 100; \\ \text{avg-queue-length} &= \text{I/O-resp} / \text{I/O-active}; \\ \text{avg-wait} &= (\text{I/O-resp} - \text{I/O-active}) / \text{I/O-ops}; \\ \text{avg-service-time} &= \text{I/O-active} / \text{I/O-ops}.\end{aligned}$$

Queue Activity

$$\begin{aligned}\text{avg-x-queue-length} &= \text{x-queue} / \text{x-queue-occupied-time}; \\ \%-\text{of-x-queue-occupied-time} &= \text{x-queue-occupied-time} / (t_2 - t_1);\end{aligned}$$

where x-queue is run queue or swap queue.

The Rest of System Activity

$$\text{avg-rate-of-x} = x / (t_2 - t_1)$$

where x is swap in/out, blks swapped in/out, terminal device activities, read/write characters, block read/write, logical read/write, process switch, system calls, read/write, fork/exec, igit, namei, directory blocks read, disk/tape I/O activities, message or semaphore activities.

If your new system will not boot, boot /ounix with the minitor. Quite likely some mistake has been made in generating the kernel. Check the date, conf.h, conf.modul, remove *.o, *.vo, *.no, *lib3 and generate the kernels again.

5.4. MUNIX/NET Update

The local area network package MUNIX/NET is closely related to MUNIX V.2 releases. This package is optional for any autonomous system, however, CADMUS 9600 systems without a disk - Diskless Nodes - require MUNIX/NET for their operation. Diskless Node systems belong to File-Servers, which may be CADMUS 9230, 9600 or 9900 systems with a disk and a filesystem for Diskless Nodes. All operations such as "removing Diskless Nodes", "installing Diskless Nodes" are executed on the File-Server. MUNIX/NET 1.3 is installed as an update to MUNIX/NET version 1.1.

The following explains the updating sequence to the network system administrator, who knows all Diskless Nodes, File-Server and autonomous systems of a network.

5.4.1. First Steps

MUNIX V.2/6a must be installed and running. Existing network information about Diskless nodes and connections to File-Server and autonomous systems may be saved.

Command	Comment
cp /etc/dllist /etc/dllist.my ll /dev/*ip* >/dev/connects.my	Diskless Nodes other systems

5.4.2. Remove MUNIX/NET 1.1

All Diskless Nodes must be inactive. Diskless Nodes on a File-Server are removed with the command rmdn. Connected systems are removed with the command rmconnect. They need not be inactive but will not be able to reach the node during the process of updating.

Command	Comment
<code>cd /usr/adm/munet</code>	
<code>rmdn</code>	repeat this command for all Diskless Nodes, see <code>/etc/dllist.my</code>
<code>rmconnect</code>	repeat this command for all connected systems, see <code>connects.my</code>

Kill all receiver und watchdog processes with `kill -9 <pids>`.

Remove the fileset MUNIX/NET 1.1. All files will be asked for separately. Please remove all of them.

Command	Comment
<code>cd / /usr/local/filesets/rmft munetlist</code>	answer y

5.4.3. Reading Fileset

This update is read like any usual fileset. Make sure you are in `/`. The fileset may follow the fileset V2-04to6a-list on the cartridge, or it may be on a separate cartridge. Please watch for the format. Both formats (ST or IS) may be read on CADMUS 9600 and 9900 models.

format	Command	Comment
ST	<code>cd / isctrl -q </dev/ris0 isskip 1 cpio -imvdS </dev/ris0</code>	after V2-04to6a-list
IS	<code>cd / </dev/ris0 cpio -imvdS </dev/ris0</code>	extra cartridge

5.4.4. Installation

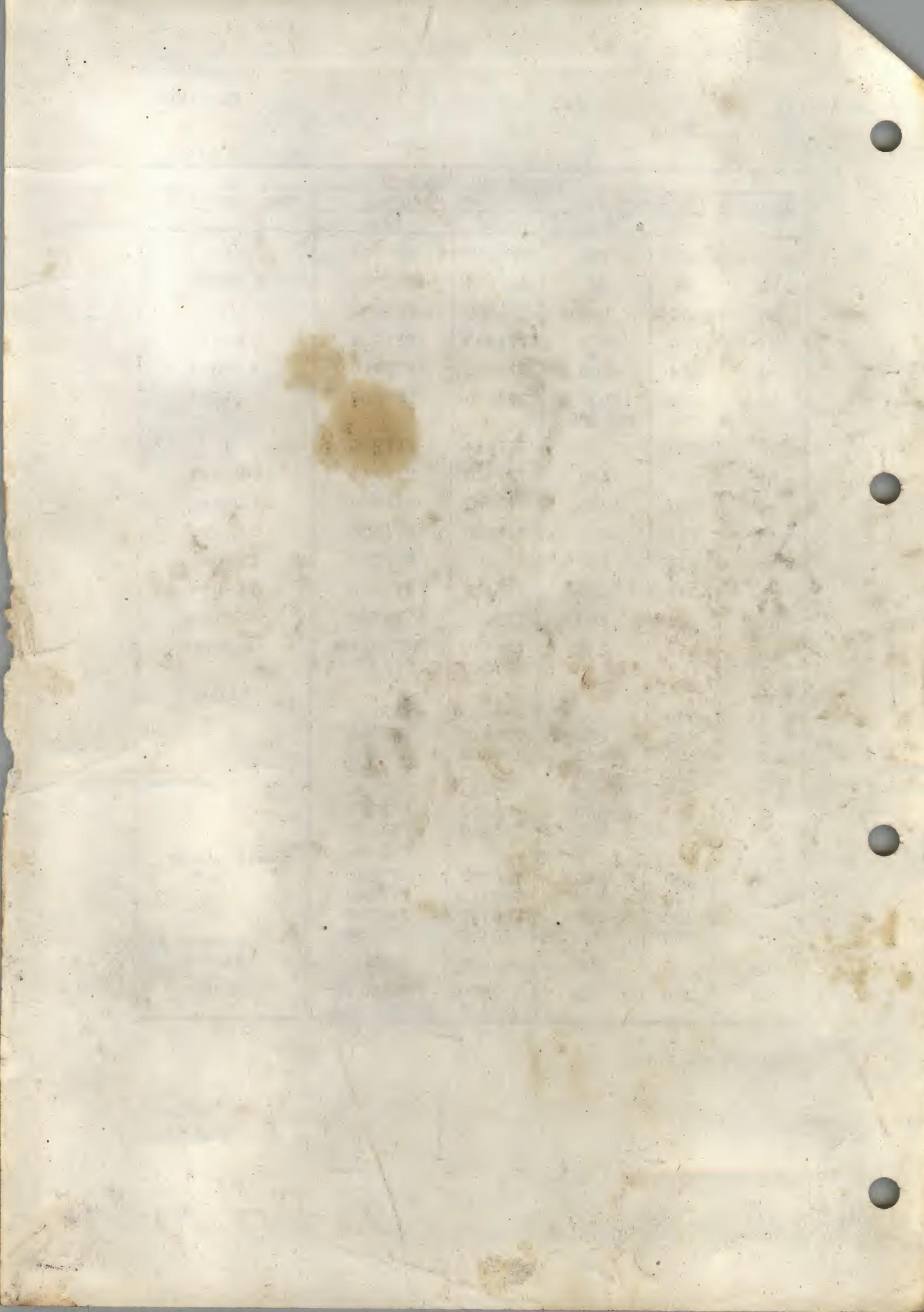
The superroot must be removed before the network software can be installed.

Diskless Nodes require their own filesystem on the File-Server. If the

Configuration Information					
Device	Interrupt Vec.		Device Address		max. Units/Lines
	octal	hex	octal	hex	
console	60	C0	777560	FFFF70	1
hb	70	E0	767770	FFEFFF8	2 Drives
lbp	100	100	770000	FFF000	1
rl	160	1C0	774400	FFF900	4 Drives
hl	164	1D0	774420	FFF910	4 Drives
vp	174	1F0 plot	777400	FFFF00	1
	204	210 print			
lp	200	200	777514	FFFF4C	1
hk	210	220	777440	FFFF20	6 Drives
rk	220	240	777400	FFFF00	4 Drives
tm	224	250	772520	FFF550	8 Drives
hp	254	2B0	776700	FFFDC0	2 Drives
rx2	264	2D0	777170	FFFE78	2 Drives
hx2	270	2E0	777150	FFFE68	2 Drives
st	270	2E0	777600	FFFF80 csr	1 Drive
			777640	FFFFA0 data	
tty					7 Lines
1	300	300	776500	FFFD40	
2	310	320	776510	FFFD48	
3	320	340	776520	FFFD50	
4	330	360	776530	FFFD58	
5	340	380	776540	FFFD60	
6	350	3A0	776550	FFFD68	
7	360	3C0	776560	FFFD70	
dz(v)					32(16) Lines
1st	330	360	760100	FFE040	
2nd	340	380	760110	FFE048	
3rd	350	3A0	760120	FFE050	
4th	360	3C0	760130	FFE058	
dh	340	380	760020	FFE010	16 Lines
td	370	3E0	777600	FFFF80 csr	2 Drives
			777640	FFFFA0 data	

FILES

/usr/sys/confinfo



File-Server is a CADMUS 9600 the filesystem for Diskless Nodes is /dev/iw0.1; other models may use /dev/hk?.? or /dev/sw0.1. Diskless Nodes are installed with the command insdn. The paper *Die Installation von MUNIX/NET* provides the system administrator with more information about the installation of File-Server and Diskless Nodes.

Connections to other systems are made with the command connectnodes.

Command	Comment
/usr/lib/munet/rejig -d	remove superroot
cd /usr/adm/munet insnetbase	install network software
/etc/mkfs /dev/riw0.1 10240 /etc/fsck /dev/riw0.1	CADMUS 9600 and 9920: Diskless Node Root-Filesystem filesystem consistency check
insdn	repeat this command for all Diskless Nodes in /etc/dllist.my
connectnodes	repeat this command for other nodes in /dev/connects.my

5.5. Finishing Touches

Go into multi users mode (**init 2**) and check that all terminals work. If everything works fine remove all saved files (i.e. all.my files, /bin/sh.sav and /etc/Init).

5.5.1. fsh

A Bourne Shell with the name fsh uses the system call forkexec. This system call is not downward compatible with any pre V.2/6a release, especially not with your ounix. After some days of satisfactory use of V.2/6a you may remove ounix and install fsh as normal shell.

Machine	Your Input
<i>INIT: SINGLE USER MODE</i>	shutdown.sh rm /ounix mv /bin/fsh /bin/sh sync init 2

What must be adapted:

File	Comment
/etc/gettydefs	if you intend to use modem control at the system console
/etc/inittab	console terminal definition add your local login terminals
/etc/passwd	add local users
/.profile	check for inconsistencies root profile
passwd root /etc/group /etc/rc	give root a password add local users add your mount commands except the mounts of diskless nodes and MUNIX/NET 1.1 software
/usr/lib/uucp/L-devices	adapt to your uucp connections with help of L-devices.my

5.2.1. Modem Control for Console

MUNIX V.2/6a gives you the possibility to use modem control at the system console, if you have the corresponding hardware, such as terminal interface (MUX-KE2, same as I/O processor) and terminal cable. To use modem control, you must delete the parameter **CLOCAL** in the console entry of /etc/gettydefs:

console entry as delivered by PCS:

```
console# B9600 HUPCL PARENBS7 OPOST ONLCR CLOCAL #
      B9600 SANE IXANY TAB3 CLOCAL #Console Login: #console
```

console entry changed to use modem control:

```
console# B9600 HUPCL PARENBS7 OPOST ONLCR #
      B9600 SANE IXANY TAB3 #Console Login: #console
```

5.2.2. Terminal Names for CADMUS 9600 and CADMUS 9900

The terminal names in /dev have been changed to correspond to the labels on the back panel of these models. If you have an I/O Processor with at least 4 serial interfaces you should make new tty devices and choose the appropriate entries in /etc/inittab.

Command	Comment
cd /dev	
make M1	for 4 serial interfaces
make M2	for 8 serial interfaces

The file **/etc/inittab** contains lines for nearly all combinations of CADMUS models and interfaces. Look for your configuration with an editor. For 4 serial interfaces and no bit-map terminal you will find the following entries:

```
# CADMUS 9600/CIP
# 1 muxke2 and 0 bitmap ==> 4 serial interfaces on the i/o processor (muxke2)
# ==> 0 serial interfaces on the icc board
# make M1
# tty11 is the console (co:)
#####t12:2:respawn: /etc/getty -h tty12 9600 none
#####t13:2:respawn: /etc/getty -h tty13 9600 none
#####t14:2:respawn: /etc/getty -h tty14 9600 none
```

Remove ##### (10*) in front of the terminal entries for the 3 terminals and change the terminal type none to pcs-qume, or your terminal type. Other combinations are handled accordingly.

5.3. New Kernel

Kernel extensions require that you generate new kernels for your system.

Use **newconf** or **genunix96** to bring your configuration files on V.2/6a level, do not edit your old configuration files.

Command	Comment
cd /usr/sys	kernels are made in this directory
/etc/newconf or genunix96	following shell procedures assist you with system configuration for all CADMUS systems
kernel.gen	system configuration for CADMUS 9600 or 9920 systems only
kernel.run	Save your old unix in ounix "install" the kernel of your choice

Autoboot the unix which you selected with **kernel.run**.